

作業二:使用mmap進行檔案複製

1.修改mycp2.c的內容:

```
inputPtr = mmap(NULL, fileSize, PROT_READ, MAP_SHARED, inputFd, 0);
ftruncate(outputFd, fileSize); // 🐻 🐱 🐭 🐹 🐰 🦊
outputPtr = mmap(NULL, fileSize, PROT_WRITE, MAP_SHARED, outputFd, 0);
```

新增這三行於初始資料的地方：

mmap()將指定檔案內容應設置記憶體中
ftruncate()讓outputFd與fileSize同大小

```
memcpy(outputPtr + data_off, inputPtr + data_off, blockSize);
```

將原mycp2複製內容的while迴圈(read&write函式的部分)改為memcpy()

於此與mmap_cp差異為mmap_cp2可支援file hole，只複製檔案中有真正內容的部分
與mycp2的差異為檔案複製改為使用mmap()、memcpy()

2.測試mmap_cp2效能與mycp2差異:

```
uneko@uneko:~/Desktop/system-programming/ch04$ time ./mmap_cp2 myHole myHole2
real    0m0.004s
user    0m0.002s
sys     0m0.000s
uneko@uneko:~/Desktop/system-programming/ch04$ time ./mycp2 myHole myHole1
real    0m0.009s
user    0m0.001s
sys     0m0.004s
```

由圖可知，

mmap_cp2所花的時間為0.004s

而mycp2所花的時間為0.009s

兩者相差0.005s

然而在測試時發現有時mmap_cp2會快一些，有時則相反，但都相差不大

```
uneko@uneko:~/Desktop/system-programming/ch04$ ./hole
這個程式會在當前的目錄下，製造檔案myHole
12K -rw----- 1 uneko uneko 4.0G 三 23 01:44 myHole
uneko@uneko:~/Desktop/system-programming/ch04$ time ./mmap_cp2 myHole myHole2

real    0m0.012s
user    0m0.004s
sys     0m0.001s
uneko@uneko:~/Desktop/system-programming/ch04$ time ./mycp2 myHole myHole1

real    0m0.011s
user    0m0.005s
sys     0m0.001s
```

之後將檔案大小調大不過mmap_cp2與mycp2複製時間依舊差異不大，因此認為兩者程式複製時間幾乎相同。